# มุม OS

# กู้ข้อมูลจากหน่วยความจำ

สุรศักดิ์ สงวนพงษ์*

*เฮ้อ...เสร็จซะที อุตส่าห์คีย์โปรแกรมเข้าไปตั้งครึ่งวัน ลองรันดูซิ...อ้าว! หลุดออกมาที่ดอส...???*

คุณผู้ใช้ไมโครฯ คงเคยพบปัญหาโปรแกรมเลิกการทำงานโดยไม่ตั้งใจ ที่สำคัญก็คือ เรายังไม่ได้เซฟข้อมูลลงดิสค์ให้เรียบร้อย ทั้งที่อุตส่าห์พิมพ์ข้อมูลเข้าไปตั้งหลายหน้ากระดาษอย่างตั้งอกตั้งใจ ผลก็เลยต้องมาเริ่มต้นนับหนึ่งพิมพ์กันใหม่อีกครั้ง

การหยุดทำงานของโปรแกรมโดยกะทันหันนี้ อาจมีสาเหตุได้หลายกรณีขึ้นกับโปรแกรมที่เราใช้อยู่ เป็นต้นว่า ใช้เวิร์ดโปรเซสเซอร์แล้วพลาดไปกดคำสั่งยกเลิกข้อมูล หรือเวิร์ดโปรเซสเซอร์บางตัว ถ้ากำลังเซฟข้อมูลแล้วพบว่าแผ่นดิสค์มีเนื้อที่เหลือไม่พอก็จะหยุดทำงานไปดื้อๆ หรือแม้กระทั่งเขียนโปรแกรมภาษาเบสิกแล้วรันโปรแกรมปรากฏว่าโปรแกรมดีบักทำให้หลุดการทำงานออกมาสู่ดอสโดยยังไม่ได้เซฟโปรแกรมเป็นต้น

มุมโอเอสในฉบับนี้จึงขอถือโอกาสเข้ามาเป็นผู้ช่วยแก้ไขปัญหานี้ คุณจะไม่ต้องมานั่งเสียเวลาพิมพ์ข้อมูลหรือโปรแกรมกันใหม่ เพียงแต่เรียนรู้เทคนิคและวิธีการเรียกเอาข้อมูลกลับคืนมา ในที่นี้จะมีข้อมูลทดลองที่ใช้เป็นตัวอย่างฝึกฝน โดยแยกปัญหากรณีเมื่อใช้เวิร์ดสตาร์ และปัญหาเมื่อเขียนโปรแกรมด้วยภาษาเบสิก

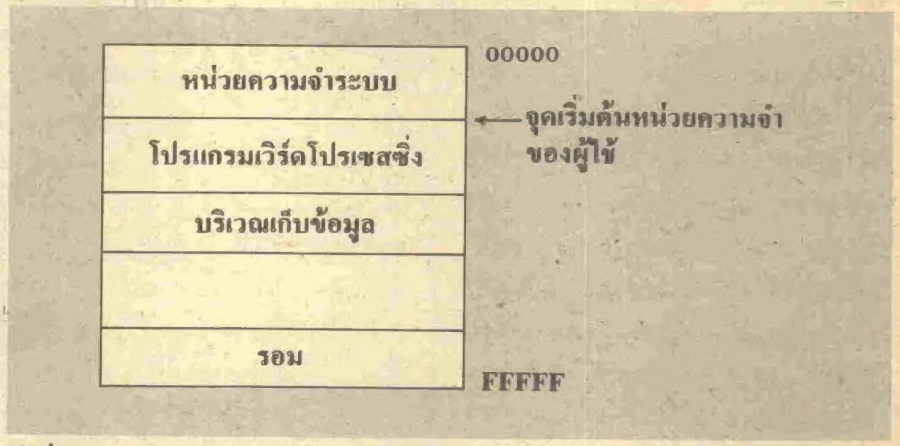
## ความเข้าใจพื้นฐาน

ก่อนที่เราจะลงมือปฏิบัติจริง ผมขอเล่าถึงหลักการคร่าว ๆ เสียก่อนว่าทำไมเราจึงยังเรียกเอาข้อมูลกลับมาได้ เมื่อเราพบปัญหาโปรแกรมหยุดการทำงานออกมาสู่ดอส และจะเรียกกลับมาด้วยวิธีใด ผมขอสมมติว่าเรากำลังใช้โปรแกรมเวิร์ดโปรเซสซิ่งอยู่ ในขณะใช้งานนั้น หน่วยความจำของเครื่องจะมีลักษณะดังรูปที่ 1

โดยปกติแล้วบริเวณที่เก็บข้อมูลนั้นจะเป็นส่วนที่อยู่ต่อจากตัวโปรแกรมเวิร์ดโปรเซสเซอร์ที่ใช้งานคือแอดเดรสที่มีค่าสูงกว่า เมื่อมีความผิดปกติเกิดขึ้นทำให้เลิกจากโปรแกรมมาอยู่ที่เครื่องหมายเตรียมพร้อมของดอส หน่วยความจำภายในเครื่องก็ยังคงอยู่อย่างปกติดังรูปที่ 1 ดังนั้นข้อมูลจึงยังคงอยู่ในหน่วยความจำ

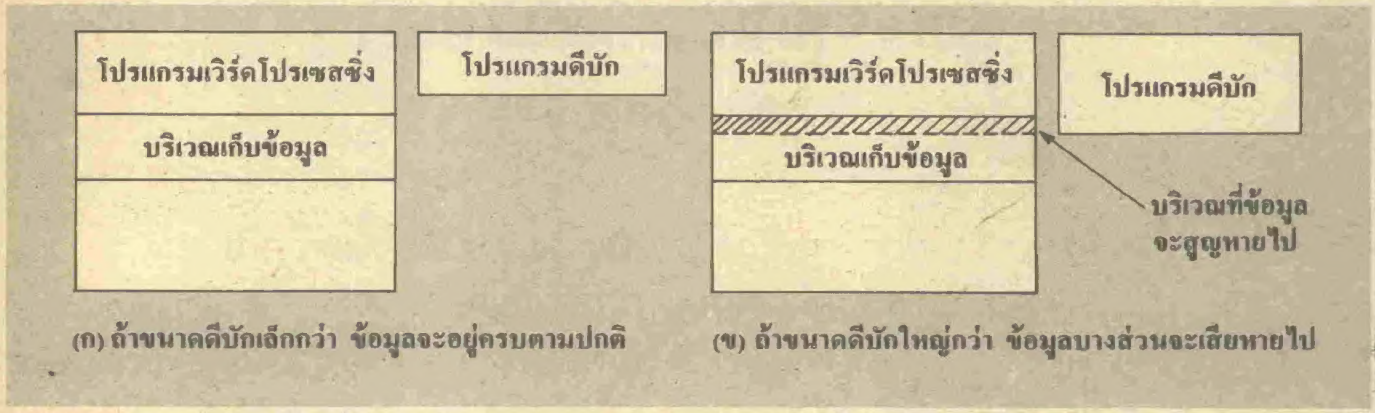
เมื่อเราทราบว่าข้อมูลของเรายังคงอยู่ในหน่วยความจำ ขั้นตอนต่อไปก็คือการดึงข้อมูลจากหน่วยความจำนั้นกลับคืนสู่ดิสค์ ซึ่งต้องใช้เครื่องมือเฉพาะงาน คือโปรแกรม DEBUG นั่นเอง เราจะโหลดดีบักลงสู่หน่วยความจำแล้วใช้ดีบักค้นหาข้อมูลให้เรา และใช้ดีบักช่วยในการเซฟข้อมูลเป็นขั้นสุดท้าย

ท่านผู้อ่านบางท่านอาจจะสงสัยว่าเมื่อตอนที่โหลดดีบักลงมานั้น ข้อมูลของเราจะไม่ถูกเขียนทับไปหรือ คำตอบก็คือมีโอกาสเป็นไปได้ครับ เพราะว่าเมื่อเราโหลดดีบักลงมานั้น ถ้าตัวดีบักมีขนาดเล็กกว่าตัวโปรแกรม ข้อมูลก็จะไม่กระทบกระเทือน แต่ถ้าดีบักมีขนาดใหญ่กว่าตัวโปรแกรมแล้ว โปรแกรมตอนท้ายของดีบักก็จะทับส่วนต้นของข้อมูล ทำให้ข้อมูลเสียไปได้ อย่างไรก็ตามเมื่อเปรียบเทียบแล้ว โปรแกรมดีบักจะมีขนาดเล็กกว่าโปรแกรมโดยทั่วไปมาก ดังนั้นโอกาสที่จะ




**รูปที่ 1 แสดงการจัดหน่วยความจำโดยคร่าว ๆ**

*ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์




(ก) ถ้าขนาดดีบักเล็กกว่า ข้อมูลจะอยู่ครบตามปกติ

(ข) ถ้าขนาดดีบักใหญ่กว่า ข้อมูลบางส่วนจะเสียหายไป

รูปที่ 2 เปรียบเทียบขนาดของดีบักกับโปรแกรม

| โปรแกรม | ขนาด (ไบท์) |
|---|---|
| DEBUG.COM | 11904 |
| DBASE.EXE | 138752 |
| BASIC.COM | 16256 |
| WS.COM | 21376 |
| BASICA.COM | 25984 |
| TURBO.COM | 39671 |

รูปที่ 3 เปรียบเทียบขนาดของดีบักกับโปรแกรมต่าง ๆ

```
a1111111111111111111111111111111111111111111111111
b2222222222222222222222222222222222222222222222222
c3333333333333333333333333333333333333333333333333
d4444444444444444444444444444444444444444444444444
e5555555555555555555555555555555555555555555555555
f6666666666666666666666666666666666666666666666666
g7777777777777777777777777777777777777777777777777
h8888888888888888888888888888888888888888888888888
i9999999999999999999999999999999999999999999999999
j0000000000000000000000000000000000000000000000000
```

รูปที่ 4 ไฟล์ TEST.TXT ใช้เป็นข้อมูลทดลอง

เกิดปัญหาการทับข้อมูลจึงมีค่อนข้างน้อย และถึงจะมีปัญหานี้เราก็ยังสามารถที่จะดึงเอาข้อมูลส่วนที่หลงเหลือกลับมาได้บ้าง รูปที่ 2 และ 3 จะช่วยชี้ให้เห็นปัญหานี้ชัดเจนยิ่งขึ้น

## กู้ข้อมูลเมื่อใช้เวิร์ดสตาร์

ตัวอย่างทดลองการกู้ข้อมูลอันแรก เราลองโดยการใช้เวิร์ดโปรเซสซิ่ง คือตัวเวิร์ดสตาร์ และเพื่อให้เห็นการทำงานได้ชัดเจนเราจะสร้างข้อมูลทดลองดังรูปที่ 4 หลังจากสร้างไฟล์ข้อมูลดังรูปที่ 4 เสร็จสิ้นแล้ว ให้ออกจากโปรแกรมเวิร์ดสตาร์ โดยกด CTRL-KX ต่อจากนั้นก็เริ่มโหลดดีบักเพื่อค้นหาข้อมูลดังรูปที่ 5

## การเซฟข้อมูลลงดิสค์

เมื่อเราค้นพบจุดเริ่มต้นของไฟล์

รูปที่ 5 การใช้ DEBUG ค้นหาข้อมูล

```
A>debug
-s0 fffe "a1"
08F1:6A18
08F1:715C
08F1:74F2
-d6a10
08F1:6A10  58 45 43 55 54 45 C9 72-61 31 31 31 31 31 31 31   XECUTElra1111111
08F1:6A20  31 31 31 31 31 31 31 31-31 31 31 31 31 31 31 31   1111111111111111
08F1:6A30  31 31 31 31 31 31 31 31-31 31 31 31 31 31 31 31   1111111111111111
08F1:6A40  31 31 31 31 31 31 31 31-31 31 0D 0A 62 32 32 32   1111111111..b222
08F1:6A50  32 32 32 32 32 32 32 32-32 32 32 32 32 32 32 32   2222222222222222
08F1:6A60  32 32 32 32 32 32 32 32-32 32 32 32 32 32 32 32   2222222222222222
08F1:6A70  32 32 32 32 32 32 32 32-32 32 32 32 32 32 0D 0A   22222222222222..
08F1:6A80  63 33 33 33 33 33 33 33-33 33 33 33 33 33 33 33   c333333333333333
-d
08F1:6A90  33 33 33 33 33 33 33 33-33 33 33 33 33 33 33 33   3333333333333333
08F1:6AA0  33 33 33 33 33 33 33 33-33 33 33 33 33 33 33 33   3333333333333333
08F1:6AB0  33 33 0D 0A 64 34 34 34-34 34 34 34 34 34 34 34   33..d44444444444
08F1:6AC0  34 34 34 34 34 34 34 34-34 34 34 34 34 34 34 34   4444444444444444
08F1:6AD0  34 34 34 34 34 34 34 34-34 34 34 34 34 34 34 34   4444444444444444
08F1:6AE0  34 34 34 34 34 34 0D 0A-65 35 35 35 35 35 35 35   444444..e5555555
08F1:6AF0  35 35 35 35 35 35 35 35-35 35 35 35 35 35 35 35   5555555555555555
08F1:6B00  35 35 35 35 35 35 35 35-35 35 35 35 35 35 35 35   5555555555555555
```

ค้นหาตำแหน่งข้อมูลในหนึ่งเซกเมนท์

แอดเดรสที่ค้นพบ ใช้เฉพาะแอดเดรสแรกเท่านั้น

ข้อมูลเริ่มต้นที่แอดเดรส 6A18

```
-d
08F1:6B10  35 35 35 35 35 35 35 35-35 35 0D 0A 66 36 36 36   5555555555..f666
08F1:6B20  36 36 36 36 36 36 36 36-36 36 36 36 36 36 36 36   6666666666666666
08F1:6B30  36 36 36 36 36 36 36 36-36 36 36 36 36 36 36 36   6666666666666666
08F1:6B40  36 36 36 36 36 36 36 36-36 36 36 36 36 36 0D 0A   66666666666666..
08F1:6B50  67 37 37 37 37 37 37 37-37 37 37 37 37 37 37 37   g7777777777777777
08F1:6B60  37 37 37 37 37 37 37 37-37 37 37 37 37 37 37 37   7777777777777777
08F1:6B70  37 37 37 37 37 37 37 37-37 37 37 37 37 37 37 37   7777777777777777
08F1:6B80  37 37 0D 0A 68 38 38 38-38 38 38 38 38 38 38 38   77..h8888888888
-d
08F1:6B90  38 38 38 38 38 38 38 38-38 38 38 38 38 38 38 38   8888888888888888
08F1:6BA0  38 38 38 38 38 38 38 38-38 38 38 38 38 38 38 38   8888888888888888
08F1:6BB0  38 38 38 38 38 38 0D 0A-69 39 39 39 39 39 39 39   888888..i9999999
08F1:6BC0  39 39 39 39 39 39 39 39-39 39 39 39 39 39 39 39   9999999999999999
08F1:6BD0  39 39 39 39 39 39 39 39-39 39 39 39 39 39 39 39   9999999999999999
08F1:6BE0  39 39 39 39 39 39 39 39-39 39 0D 0A 6A 30 30 30   9999999999..j000
08F1:6BF0  30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30   0000000000000000
08F1:6C00  30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30   0000000000000000
-d
08F1:6C10  30 30 30 30 30 30 30 30-30 30 30 30 30 30 0D 0A   00000000000000..
08F1:6C20  1A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C30  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C40  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C50  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C60  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C70  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
08F1:6C80  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

จุดสุดท้ายของไฟล์ซึ่งปิดท้ายด้วยรหัส 1A ที่แอดเดรส 6C20

```
-h 6c20 6a18      - ใช้ดีบักคำนวณขนาดไฟล์
D638  0208        - ขนาดไฟล์ 208 ไบท์
-rbx              - เซ็ท BX ให้เป็น 0
BX 0000
:
-rcx              - เซ็ท CX ให้เท่ากับขนาดไฟล์คือ 208
CX 0000
:208
-r                - ตรวจรีจิสเตอร์
AX=0000  BX=0000  CX=0208  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=08F1  ES=08F1  SS=08F1  CS=08F1  IP=0100   NV UP DI PL NZ NA PO NC
08F1:0100 C9            DB      C9
-n test.sav       - ตั้งชื่อไฟล์
-w 6a18           - เซฟลงดิสค์
Writing 0208 bytes
-q

A>dir test.sav

 Volume in drive A has no label
 Directory of  A:\                    - ไฟล์ที่เซฟได้

TEST     SAV       520   1-01-80  12:31a
        1 File(s)     41984 bytes free
```

รูปที่ 6 การเซฟข้อมูลด้วยดีบัก

ข้อมูล เราก็จำเป็นต้องหาจุดสุดท้ายของข้อมูลซึ่งสำหรับไฟล์ข้อมูลแบบอัสกีแล้วจะสังเกตได้ว่ามีรหัส 1A (CTRL-Z) เป็นตัวบอกจุดสิ้นสุดไฟล์ ในกรณีนี้ก็คือแอดเดรส 6C20 จากนั้นก็ใช้ดีบักคำนวณขนาดไฟล์แล้วใส่คำไว้ในรีจิสเตอร์ CX เพื่อเซฟไฟล์ดังรูปที่ 6

เนื่องจากว่ากรณีนี้เป็นการทดลอง เราจึงมองเห็นรหัส 1A เป็นตัวบอกจุดสิ้นสุดของไฟล์ แต่ปัญหาจริงนั้น เราอาจจะไม่พบรหัสนี้ ดังนั้นการหาจุดสิ้นสุดของข้อมูลจำเป็นต้องมองหาเองว่าข้อความนั้นสิ้นสุดที่จุดใด โดยสังเกตจากข้อความทางขวามือของดีบัก ซึ่งแสดงอยู่ในรูปรหัสอัสกีเป็นตัวอักษรที่อ่านได้

อย่างไรก็ตามยังมีปัญหาอีกกรณีที่น่าคำนึงถึงคือ ถ้าเราพบข้อมูลเพียงบางส่วนในหน่วยความจำทั้งที่แน่ใจว่าดีบักไม่ได้ทับข้อมูลส่วนที่หายไป ปัญหานี้เกิดจากไฟล์ข้อมูลของเราอาจจะใหญ่โตเกินกว่าที่โปรแกรมจะนำลงมาไว้ในหน่วยความจำทั้งหมด ดังนั้นข้อมูลส่วนที่หายไปก็คงยังอยู่บนดิสค์นั่นเอง โดยปกติแล้วเมื่อเราสร้างไฟล์สมมติว่าชื่อ SAMPLE.TXT เมื่อเรียกไฟล์นี้มาแก้ไข ตัวเวิร์ดโปรเซสซิ่งจะสร้างไฟล์ ๆ ใหม่ที่มีหน้าตาเหมือน SAMPLE.TXT ทุกประการแต่ใช้เป็นชื่อ SAMPLE.$$$ และใช้ไฟล์นามสกุล $$$ นี้ให้เราแก้ไข ในกรณีที่แก้ไขเสร็จสิ้นแล้วก็จะมีการเปลี่ยนชื่อ SAMPLE.TXT ไปเป็น SAMPLE.BAK และเปลี่ยน SAMPLE.$$$ ไปเป็น SAMPLE.TXT แทน แต่ถ้าเรายกเลิกการแก้ไข ไฟล์ SAMPLE.$$$ จะลบทิ้งไปโดยปริยาย ดังนั้นเราจึงไม่เห็นไฟล์ $$$ อยู่บนแผ่นดิสค์เลย แต่ในกรณีที่มีการผิดพลาดของโปรแกรมเราจะพบเห็นไฟล์ $$$ อยู่ และวิธีการกู้ข้อมูลก็ทำได้โดยเซฟข้อมูลในหน่วยความจำแล้วนำมาตัดต่อเข้ากับไฟล์ $$$

```
A>debug
-d 0050:0010 12              - ขอดูเซกเมนท์โปรแกรม 2 ไบท์
0050:0010  0A 0B
-d 0b0a:30 12                - ที่ออฟเซ็ท 30 จะเป็นแอดเดรสเริ่มต้น
0B0A:0030  87 0E
-f 0b0a:0e87 11 ff           - ใส่ค่า FF ที่แอดเดรสเริ่มต้นนี้

-d0b0a:358 12                - ที่ออฟเซ็ท 358 เป็นแอดเดรสสุดท้าย
0B0A:0358  FC 14
-h 14fc 0e87                 - คำนวณขนาดโปรแกรม
2383  0675
-rcx                         - เซ็ทรีจิสเตอร์ CX ให้เท่ากับขนาดโปรแกรม
CX 0000
:675
-rbx
BX 0000
:
-n test.bas                  - ตั้งชื่อไฟล์
-w 0b0a:0e87                 - เซฟลงดิสก์
Writing 0675 bytes
-q

A>
```

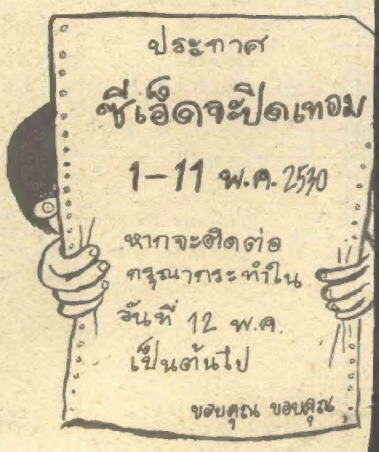
รูปที่ 7 การเซฟโปรแกรมเบสิก

## กู้ข้อมูลเมื่อเขียนโปรแกรมเบสิก

ในกรณีของโปรแกรมภาษาเบสิกนั้น การดูข้อมูลในหน่วยความจำค่อนข้างจะเป็นเรื่องยุ่งยาก เนื่องจากการเก็บโปรแกรมจะเก็บเป็นรหัสโทเกน การค้นหาตำแหน่งข้อมูลจึงเป็นเรื่องยาก อย่างไรก็ตามแทนที่จะใช้วิธีการค้นดูข้อมูล เราจะใช้ดีบักเข้าไปตรวจสอบจุดเริ่มและจุดสิ้นสุด โดยดูที่บริเวณหน่วยความจำที่เก็บข้อมูลนี้อยู่แล้ว จึงเซฟโปรแกรมลงสู่ดิสก์ดังรูปที่ 7

ในขั้นแรกนั้นให้ตรวจดูว่าเซกเมนท์และแอดเดรสที่ใช้เก็บโปรแกรมนั้น มีค่าเท่าไรโดยดูจากหน่วยความจำที่แอดเดรส 0050:0010 จากนั้นก็นำแอดเดรสนี้มาตรวจดูว่าจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรมอยู่ที่ใด โดยดูจากแอดเดรสออฟเซ็ท 30 และ 358 ตามลำดับ จากนั้นจึงใช้ดีบักคำนวณหาขนาด แล้วเซฟโปรแกรมเป็นอันเสร็จวิธี

## บทสรุป

การกู้ข้อมูลจากหน่วยความจำที่กล่าวมาทั้งหมดนี้ จะทำได้ในกรณีเฉพาะที่โปรแกรมที่ใช้งานเกิดการผิดพลาดทำให้หลุดมาอยู่ที่ดอส แต่ในบางกรณีที่โปรแกรมมีปัญหาแล้ว เครื่องเกิดอาการแฮงก็คงไม่มีวิธีช่วยเหลืออย่างอื่น นอกจากรีเซ็ทระบบใหม่อีกครั้ง

ตัวอย่างที่ยกมานี้ สามารถนำไปประยุกต์ใช้กับโปรแกรมอื่น ๆ ได้ เพราะยังคงใช้หลักการเช่นเดียวกัน ที่สำคัญก็คือลองฝึกฝนให้ชำนาญแล้วจะพบว่าตัวท่านเองก็เป็นนักกู้ข้อมูลมือฉมังอีกผู้หนึ่ง